Unconditionally Energy Stable Immersed Boundary Method with Application to Vesicle Dynamics

Wei-Fan Hu and Ming-Chih Lai*

Center of Mathematical Modeling and Scientific Computing & Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan.

Abstract. We develop an unconditionally energy stable immersed boundary method, and apply it to simulate 2D vesicle dynamics. We adopt a semi-implicit boundary forcing approach, where the stretching factor used in the forcing term can be computed from the derived evolutional equation. By using the projection method to solve the fluid equations, the pressure is decoupled and we have a symmetric positive definite system that can be solved efficiently. The method can be shown to be unconditionally stable, in the sense that the total energy is decreasing. A resulting modification benefits from this improved numerical stability, as the time step size can be significantly increased (the severe time step restriction in an explicit boundary forcing scheme is avoided). As an application, we use our scheme to simulate vesicle dynamics in Navier-Stokes flow.

1. Introduction

The immersed boundary (IB) method proposed by Peskin [26] has been successfully applied to many fluid-structure interaction problems — cf. the review [27]. The IB method employs an Eulerian description for the fluid velocity and a Lagrangian description for the configuration of the immersed elastic structure (immersed boundary or interface). The immersed structure exerts some force into the fluid that drives the fluid flow, and at the same time the fluid flow carries the immersed structure to a new configuration. This interaction between the fluid and the immersed structure is linked through a force spreading and velocity interpolating operator, on using a smoothed version of the Dirac delta function [27]. The IB method is easy to implement and efficient, simply because the immersed structure

---

*Corresponding author. *Email addresses:* `weifanhu.am95g@g2.nctu.edu.tw` (W.-F. Hu), `mclai@math.nctu.edu.tw` (M.-C. Lai)

(no matter how complex) is regarded as a force generator to the fluid, so that the fluid variables can be solved in a fixed Eulerian domain without generating any structure-fitting grid. Many fast efficient fluid solvers can therefore be applied.

Despite substantial success with practical applications using the IB method, it still has some drawbacks from the numerical point of view. Firstly, the method is only first-order accurate, whereas second-order accurate fluid solvers are used. The immersed elastic structure is usually represented one-dimensionally lower than the fluid space so that the exerted force is singular (delta function like), and smoothing the delta function in a regular finite difference scheme causes the method to be first-order accurate only. Although there have been several attempts to improve accuracy, even some including adaptive local mesh refinements near the immersed boundary, formally those methods still remain to be made second-order accurate [6, 7, 17, 22, 28].

Another issue is numerical stability. As is well known, the IB method suffers a time step restriction to maintain numerical stability [21, 27, 29, 30]. This restriction becomes more stringent when the elastic force is stiff and the force spreading occurs at the beginning of each time step (an explicit scheme). It is notable that such a time step restriction cannot be alleviated even when the fluid solver is discretised in a semi-implicit manner — i.e. with explicit differencing of the advection term and implicit differencing of the diffusion term. Rather than performing the force spreading at the beginning of the time step, one might consider doing so at an intermediate stage (a semi-implicit scheme) or even at the end of the time step (an implicit scheme). In the past decade, there have been many attempts to reduce the stiffness or to overcome this time step restriction [3, 4, 8, 10, 11, 23, 24]. However, there is always a trade-off between the stability and efficiency of the algorithms involved. In this article, we propose a new semi-implicit scheme that can be solved quite efficiently, where the resultant linear system is symmetric positive definite and the time step size can be significantly increased.

In Section 2, we introduce the formulation for the incompressible Navier-Stokes equations with an immersed elastic interface. We then develop semi-implicit immersed boundary schemes based on the projection method for the fluid solver in Section 3, and show that these developed schemes are unconditionally energy stable. Then we modify these semi-implicit schemes for efficient implementation, with the resultant linear system symmetric positive definite. Numerical results from simulations of vesicle dynamics are given in Section 4, followed by our conclusions and discussion of future work in Section 5.

2. Governing Equations

We begin by stating the mathematical formulation of the Navier-Stokes flow with an immersed boundary (or interface). We consider a moving, immersed, elastic boundary $\Gamma(t)$, which exerts forces into an incompressible fluid in a fixed fluid domain Ω. We assume that the fluids inside and outside of the boundary are the same, so the governing equations can be written as follows:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u} + \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) \, \mathrm{d}s \quad \text{in } \Omega \,, \qquad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0 \ \text{ in } \Omega \,, \tag{2.2}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) = \int_{\Omega} \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t)) \, \mathrm{d}\mathbf{x} \ \text{ on } \Gamma \,. \tag{2.3}$$

Here ρ and μ are constant fluid density and viscosity, $\mathbf{u} = (u,v)$ and p are the velocity field and the pressure, both described in Cartesian coordinates. The immersed boundary Γ is represented by a parametric form $\mathbf{X}(s,t) = (X(s,t), Y(s,t))$, where s is the Lagrangian parameter of the initial configuration. The unit tangent vector $\boldsymbol{\tau}$ is defined as $\boldsymbol{\tau} = \mathbf{X}_s/|\mathbf{X}_s|$.

Eqs. (2.1) and (2.2) are the familiar incompressible Navier-Stokes equations, involving a force from the immersed boundary Γ represented by the line integral. Eq. (2.3) defines the position of the immersed boundary moving with the local fluid velocity (the interfacial velocity). Thus the interfacial velocity $\mathbf{U}$ is simply an interpolation of the fluid velocity at the immersed boundary, and the interaction between the fluid and the immersed boundary is linked by the two-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)$. We envisage that the force arises from elastic tension according to Hooke's law — i.e.

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \quad \text{where} \quad \sigma = \sigma_0(|\mathbf{X}_s| - r_0) \,, \tag{2.4}$$

with σ_0 is the elastic coefficient and r_0 the rest length. Note that the rate of local stretching can then be considered to satisfy the following equation [18]:

$$\frac{\partial}{\partial t}|\mathbf{X}_s| = (\nabla_s \cdot \mathbf{U}) \, |\mathbf{X}_s| = \frac{\partial \mathbf{U}}{\partial s} \cdot \boldsymbol{\tau} \,. \tag{2.5}$$

3. Numerical Discretization

We now proceed to discretise Eqs. (2.1)–(2.5) by the IB method, in a computational rectangular domain $\Omega = [a,b] \times [c,d]$ for simplicity. The fluid variables are defined on the staggered marker-and-cell (MAC) grid introduced by Harlow & Welsh [9] — i.e. the pressure is defined at the cell centre labelled

$$\mathbf{x} = (x_i, y_j) = (a + (i - 1/2)\Delta x, c + (j - 1/2)\Delta y) \,, \ i = 1, 2, \cdots, m \,, \ j = 1, 2, \cdots, n \,,$$

while the velocity components u and v are defined at the cell edges

$$(x_{i-1/2}, y_j) = (a + (i-1)\Delta x, c + (j-1/2)\Delta y) \,, \ (x_i, y_{j-1/2}) = (a + (i-1/2)\Delta x, c + (j-1)\Delta y) \,.$$

We assume a uniform mesh width $h = \Delta x = \Delta y$ in the fluid, although that is not necessary. For the immersed boundary, we use a collection of discrete points $s_k = k\Delta s$, $k = 0, 1, \cdots M$ where the mesh width Δs is chosen to be roughly half of the fluid mesh width h. The Lagrangian markers are denoted by $\mathbf{X}_k = \mathbf{X}(s_k)$. We also assume the immersed boundary is closed, so that $\mathbf{X}_0 = \mathbf{X}_M$. The elastic tension is written $\sigma_{k-1/2}$, since it is defined at the "half-integer" points $s_{k-1/2} = (k - 1/2)\Delta s$. Without loss of generality, for any function
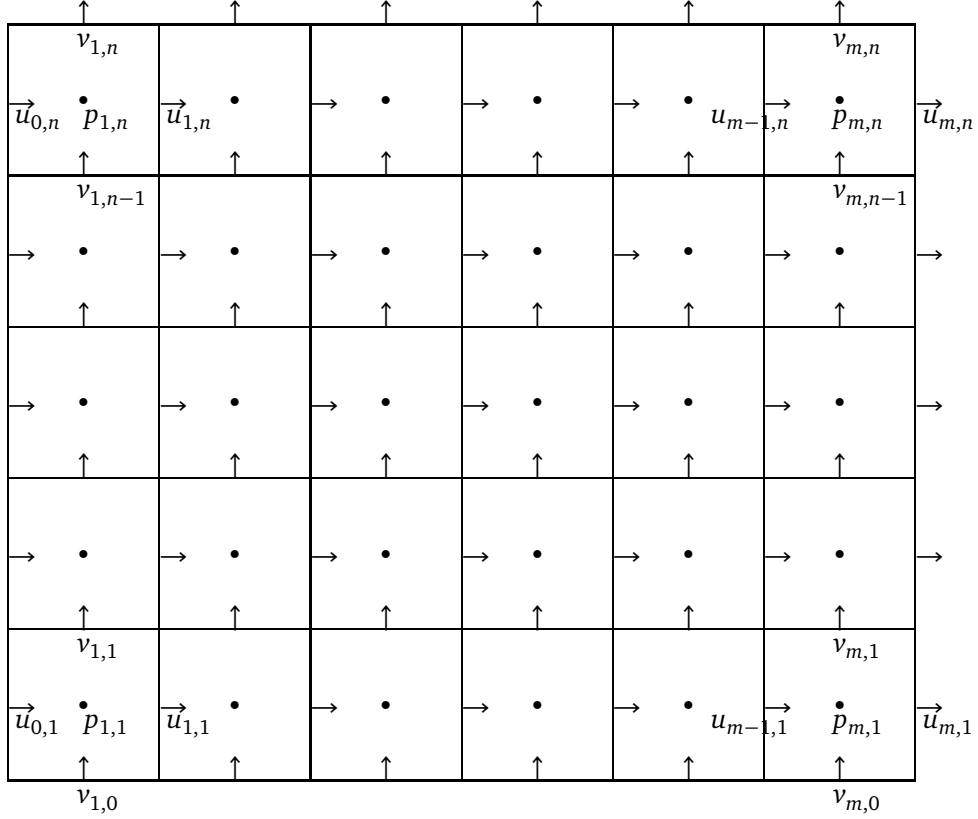
Figure 1: The computational domain Ω_h with a staggered grid.

$\phi(s)$ defined on the immersed boundary, we approximate the partial derivative $\partial \phi / \partial s$ by central differences — i.e.

$$D_s \phi = \frac{\phi(s + \Delta s/2) - \phi(s - \Delta s/2)}{\Delta s} \, .$$

The unit tangent approximated by $\tau = D_s \mathbf{X} / |D_s \mathbf{X}|$ is consequently defined at the half-integer points.

Let Δt be the time step size, and the superscript index denote the time step level. At the beginning of each time level n, the values of the boundary configuration $\mathbf{X}_k^n$ and the unit tangent $\tau_{k-1/2}^n$ are all given. To simplify our notation, we define the discrete spreading operator acting on the tension and discrete surface divergence operator acting on the velocity as follows:

$$\mathscr{S}_h^m[\sigma](\mathbf{x}) = \mathscr{S}_h^m[\sigma_0(|\mathbf{X}_s| - r_0)](\mathbf{x}) = \sum_{k=1}^{M} D_s \left(\sigma_0(|\mathbf{X}_s| - r_0)\tau^m \right)_k \delta_h(\mathbf{x} - \mathbf{X}_k^m) \Delta s , \quad (3.1)$$

$$\mathscr{T}_h^m[\mathbf{u}](s_{k-1/2}) = \frac{\mathbf{U}_k - \mathbf{U}_{k-1}}{\Delta s} \cdot \tau_{k-1/2}^m \quad \mathbf{U}_k = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}_k^m)h^2 , \quad\quad (3.2)$$

where the superscript m denotes values for the immersed boundary position at the time level $m\Delta t$ and the discrete operators are skew-adjoint as $\langle \mathscr{S}_h^m[\sigma], \mathbf{u}\rangle_{\Omega_h} = \langle \sigma, -\mathscr{T}_h^m[\mathbf{u}]\rangle_{\Gamma_h}$ — cf. the derivation in Ref. [19]. The skew-adjointness property plays an important role in our energy stability analysis later.

Our numerical discretisation and energy stability analysis are based on unsteady Stokes equations in this article. The nonlinear advection term in the Navier-Stokes equation (2.1) can be treated explicitly during the time evolution with a moderate CFL condition. However, the Navier-Stokes equations can also be split into an advection part and unsteady Stokes part, where the advection part can be solved by an alternating direction implicit (ADI) method to maintain the unconditionally numerical stability [11].

We introduce two time integrations, the backward Euler (BE) and Crank-Nicholson (CN) scheme, in conjunction with the projection method as follows [5]:

- Backward Euler (BE) scheme

$$\rho \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mu \Delta_h \mathbf{u}^* + \mathscr{S}_h^n[\sigma_0(|\mathbf{X}_s|^{n+1} - r_0)] , \ \text{in } \Omega_h ; \tag{3.3}$$

$$\Delta_h \phi = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^* , \ \text{in } \Omega_h ; \ \left.\frac{\partial \phi}{\partial \mathbf{n}}\right|_{\partial \Omega_h} = 0 ; \tag{3.4}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi , \ \text{in } \Omega_h ; \tag{3.5}$$

$$\nabla_h p^{n+1} = \rho \nabla_h \phi - \mu \nabla_h(\nabla_h \cdot \mathbf{u}^*) , \ \text{in } \Omega_h ; \tag{3.6}$$

$$\frac{|\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n}{\Delta t} = \mathscr{T}_h^n[\mathbf{u}^{n+1}] , \ \text{on } \Gamma_h ; \tag{3.7}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})^{n+1} \delta_h(\mathbf{x} - \mathbf{X}_k^n) h^2 ; \tag{3.8}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t \, \mathbf{U}_k^{n+1} . \tag{3.9}$$

- Crank-Nicholson (CN) scheme

$$\rho \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\mu}{2} \Delta_h(\mathbf{u}^* + \mathbf{u}^n) + \mathscr{S}_h^n\left[\sigma_0\left(\frac{|\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^n}{2} - r_0\right)\right] , \ \text{in } \Omega_h ; \tag{3.10}$$

$$\Delta_h \phi = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^* , \ \text{in } \Omega_h ; \ \left.\frac{\partial \phi}{\partial \mathbf{n}}\right|_{\partial \Omega_h} = 0 ; \tag{3.11}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi , \ \text{in } \Omega_h ; \tag{3.12}$$

$$\nabla_h p^{n+1/2} = \rho \nabla_h \phi - \mu \nabla_h(\nabla_h \cdot \mathbf{u}^*) , \ \text{in } \Omega_h ; \tag{3.13}$$

$$\frac{|\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n}{\Delta t} = \mathscr{T}_h^n[(\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n)/2] , \ \text{on } \Gamma_h ; \tag{3.14}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})^{n+1} \delta_h(\mathbf{x} - \mathbf{X}_k^n) h^2 , \quad \widehat{\mathbf{U}}_k^n = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})^n \delta_h(\mathbf{x} - \mathbf{X}_k^n) h^2 ; \tag{3.15}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t(\mathbf{U}_k^{n+1} + \widehat{\mathbf{U}}_k^n)/2 . \tag{3.16}$$

The gradient and Laplacian spatial operators ∇_h and Δ_h above are represented in terms of the standard second-order central difference approximations, and δ_h is a smoother version of the discrete delta function developed in Ref. [32]. Unlike the explicit scheme, both of these backward Euler and Crank-Nicholson schemes treat the stretching factor $|\mathbf{X}_s|^{n+1}$ as an unknown solution with closure via the discretisation of Eq. (2.5), so in the immersed boundary force computation the two schemes are semi-implicit.

3.1. Energy stability analysis

We now perform the energy stability analysis for our present schemes in a similar fashion to Refs. [11, 23], and show that both methods are unconditionally stable in the sense that the total energy decreases. To proceed, we define the kinetic energy K and potential energy P of the system as follows:

$$K = \frac{\rho}{2}\|\mathbf{u}\|_{\Omega_h}^2 = \frac{\rho}{2}\langle \mathbf{u}, \mathbf{u} \rangle_{\Omega_h}, \quad P = \frac{\sigma_0}{2}\||\mathbf{X}_s| - r_0\|_{\Gamma_h}^2 = \frac{\sigma_0}{2}\langle |\mathbf{X}_s| - r_0, |\mathbf{X}_s| - r_0 \rangle_{\Gamma_h}, \quad (3.17)$$

where the associated discrete inner products are defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\Omega_h} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) h^2, \qquad \langle \phi, \psi \rangle_{\Gamma_h} = \sum_{k=1}^{M} \phi_{k-1/2}\, \psi_{k-1/2} \Delta s, \qquad (3.18)$$

respectively. The total energy is $E = K + P$.

Theorem 3.1. *The backward Euler (BE) scheme of Eqs.* (3.3)–(3.9) *is unconditionally energy stable — i.e. the scheme satisfies $E^{n+1} \le E^n$ for each time step n.*

Proof. We first consider stability analysis for the BE scheme. For convenience, let us denote the singular force term in Eq. (3.3) by $\mathscr{S}_h^n[\sigma^{n+1}]$. We first substitute $\mathbf{u}^*$ from Eq. (3.5) into Eq. (3.3) to get the equation for $\rho(\mathbf{u}^{n+1} - \mathbf{u}^n)$, and then take the discrete inner product with $\mathbf{u}^{n+1} + \mathbf{u}^n$ to obtain the kinetic energy difference

$$K^{n+1} - K^n = \frac{\rho}{2}\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1} \rangle_{\Omega_h} - \frac{\rho}{2}\langle \mathbf{u}^n, \mathbf{u}^n \rangle_{\Omega_h}$$

$$= \frac{\rho}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n \rangle_{\Omega_h}$$

$$= \frac{\rho}{2}\left(-\langle \mathbf{u}^{n+1} - \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n \rangle_{\Omega_h} + 2\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1} - \mathbf{u}^n \rangle_{\Omega_h} \right)$$

$$= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{\Omega_h}^2 + \langle \mathbf{u}^{n+1}, \rho\left(\mathbf{u}^{n+1} - \mathbf{u}^n\right) \rangle_{\Omega_h}$$

$$= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{\Omega_h}^2 + \Delta t\langle \mathbf{u}^{n+1}, -\rho\nabla_h\phi + \mu\Delta_h\mathbf{u}^{n+1} + \mu\Delta t\nabla_h\Delta_h\phi + \mathscr{S}_h^n[\sigma^{n+1}] \rangle_{\Omega_h}$$

$$= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{\Omega_h}^2 - \Delta t\rho\langle \mathbf{u}^{n+1}, \nabla_h\phi \rangle_{\Omega_h} + \mu\Delta t\langle \mathbf{u}^{n+1}, \Delta_h\mathbf{u}^{n+1} \rangle_{\Omega_h}$$

$$+ \mu\Delta t^2\langle \mathbf{u}^{n+1}, \nabla_h\Delta_h\phi \rangle_{\Omega_h} + \Delta t\langle \mathbf{u}^{n+1}, \mathscr{S}_h^n[\sigma^{n+1}] \rangle_{\Omega_h}.$$

The potential energy difference is

$$
\begin{aligned}
P^{n+1} - P^n &= \frac{\sigma_0}{2} \left(\langle |\mathbf{X}_s|^{n+1} - r_0, |\mathbf{X}_s|^{n+1} - r_0 \rangle_{\Gamma_h} - \langle |\mathbf{X}_s|^n - r_0, |\mathbf{X}_s|^n - r_0 \rangle_{\Gamma_h} \right) \\
&= \frac{\sigma_0}{2} \langle |\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^n - 2r_0, |\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n \rangle_{\Gamma_h} \\
&= \frac{\sigma_0}{2} \langle 2|\mathbf{X}_s|^{n+1} - 2r_0 - \Delta t \mathscr{T}_h^n[\mathbf{u}^{n+1}], \Delta t \mathscr{T}_h^n[\mathbf{u}^{n+1}] \rangle_{\Gamma_h} \quad \left(\text{as } |\mathbf{X}_s|^n = |\mathbf{X}_s|^{n+1} - \Delta t \mathscr{T}_h^n[\mathbf{u}^{n+1}] \right) \\
&= \langle \sigma_0(|\mathbf{X}_s|^{n+1} - r_0) - \frac{\sigma_0 \Delta t}{2} \mathscr{T}_h^n[\mathbf{u}^{n+1}], \Delta t \mathscr{T}_h^n[\mathbf{u}^{n+1}] \rangle_{\Gamma_h} \\
&= \Delta t \langle \sigma^{n+1}, \mathscr{T}_h^n[\mathbf{u}^{n+1}] \rangle_{\Gamma_h} - \frac{\sigma_0 \Delta t^2}{2} \| \mathscr{T}_h^n[\mathbf{u}^{n+1}] \|_{\Gamma_h}^2 .
\end{aligned}
$$

Thus the total energy difference between two successive time steps is

$$
\begin{aligned}
&E^{n+1} - E^n \\
&= -\frac{\rho}{2} \| \mathbf{u}^{n+1} - \mathbf{u}^n \|_{\Omega_h}^2 - \Delta t \rho \langle \mathbf{u}^{n+1}, \nabla_h \phi \rangle_{\Omega_h} + \mu \Delta t \langle \mathbf{u}^{n+1}, \Delta_h \mathbf{u}^{n+1} \rangle_{\Omega_h} + \mu \Delta t^2 \langle \mathbf{u}^{n+1}, \nabla_h \Delta_h \phi \rangle_{\Omega_h} \\
&\quad + \Delta t \langle \mathscr{S}_h^n[\sigma^{n+1}], \mathbf{u}^{n+1} \rangle_{\Omega_h} + \Delta t \langle \sigma^{n+1}, \mathscr{T}_h^n[\mathbf{u}^{n+1}] \rangle_{\Gamma_h} - \frac{\sigma_0 \Delta t^2}{2} \| \mathscr{T}_h^n[\mathbf{u}^{n+1}] \|_{\Gamma_h}^2 .
\end{aligned}
$$

The second and fourth terms vanish due to the orthogonality of the discrete divergence-free velocity and the gradient, and the commutativity of ∇_h and Δ_h. The fifth and sixth terms cancel out since both discrete operators are skew-adjoint as $\langle \mathscr{S}_h^n[\sigma], \mathbf{u} \rangle_{\Omega_h} = \langle \sigma, -\mathscr{T}_h^n[\mathbf{u}] \rangle_{\Gamma_h}$. The third term is always negative since the Laplace operator is negative definite. Thus

$$
E^{n+1} - E^n = -\frac{\rho}{2} \| \mathbf{u}^{n+1} - \mathbf{u}^n \|_{\Omega_h}^2 - \mu \Delta t \| \nabla_h \mathbf{u}^{n+1} \|_{\Omega_h}^2 - \frac{\sigma_0 \Delta t^2}{2} \| \mathscr{T}_h^n[\mathbf{u}^{n+1}] \|_{\Gamma_h}^2 ,
$$

so the total energy is decreasing such that the present BE scheme is unconditionally energy stable. □

Theorem 3.2. *The Crank-Nicholson (CN) scheme of Eqs. (3.10)-(3.16) is unconditionally energy stable — i.e. the scheme satisfies $E^{n+1} \leq E^n$ for each time step n.*

Proof. The proof is similar to that for Theorem 3.1. Let us again denote the singular force term in Eq. (3.10) by $\mathscr{S}_h^n[(\sigma^{n+1} + \sigma^n)/2]$, substitute $\mathbf{u}^*$ in Eq. (3.12) into Eq. (3.10) to get $\rho(\mathbf{u}^{n+1} - \mathbf{u}^n)$, and then take the discrete inner product with $\mathbf{u}^{n+1} + \mathbf{u}^n$ to obtain the kinetic energy difference

$$
\begin{aligned}
K^{n+1} - K^n &= \frac{\rho}{2} \langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1} \rangle_{\Omega_h} - \frac{\rho}{2} \langle \mathbf{u}^n, \mathbf{u}^n \rangle_{\Omega_h} \\
&= \frac{\rho}{2} \langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n \rangle_{\Omega_h} \\
&= \frac{\Delta t}{2} \langle \mathbf{u}^{n+1} + \mathbf{u}^n, -\rho \nabla_h \phi + \frac{\mu}{2} \Delta_h (\mathbf{u}^{n+1} + \mathbf{u}^n) + \frac{\mu \Delta t}{2} \nabla_h \Delta_h \phi + \mathscr{S}_h^n[(\sigma^{n+1} + \sigma^n)/2] \rangle_{\Omega_h} .
\end{aligned}
$$

The potential energy difference is

$$
\begin{aligned}
P^{n+1} - P^n &= \frac{\sigma_0}{2}\left(\langle |\mathbf{X}_s|^{n+1} - r_0, |\mathbf{X}_s|^{n+1} - r_0\rangle_{\Gamma_h} - \langle |\mathbf{X}_s|^n - r_0, |\mathbf{X}_s|^n - r_0\rangle_{\Gamma_h}\right)\\
&= \frac{\sigma_0}{2}\langle |\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^n - 2r_0, |\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n\rangle_{\Gamma_h}\\
&= \frac{\Delta t}{4}\langle \sigma_0(|\mathbf{X}_s|^{n+1} - r_0 + |\mathbf{X}_s|^n - r_0), \mathscr{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h} \ \text{(using Eq. (3.14))}\\
&= \frac{\Delta t}{4}\langle \sigma^{n+1} + \sigma^n, \mathscr{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h}.
\end{aligned}
$$

Thus the total energy difference between two successive time steps is

$$
\begin{aligned}
E^{n+1} - E^n &= \frac{-\rho\Delta t}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \nabla_h\phi\rangle_{\Omega_h} + \frac{\mu\Delta t}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \Delta_h(\mathbf{u}^{n+1} + \mathbf{u}^n)\rangle_{\Omega_h}\\
&\quad + \frac{\mu\Delta t^2}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \nabla_h\Delta_h\phi\rangle_{\Omega_h} + \frac{\Delta t}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathscr{S}_h^n[\sigma^{n+1} + \sigma^n]\rangle_{\Omega_h}\\
&\quad + \frac{\Delta t}{4}\langle \sigma^{n+1} + \sigma^n, \mathscr{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h} = -\frac{\mu\Delta t}{4}\|\nabla_h(\mathbf{u}^{n+1} + \mathbf{u}^n)\|^2_{\Omega_h},
\end{aligned}
$$

The first and third terms vanish due to the orthogonality of discrete divergence-free velocity and the gradient, and the commutativity of ∇_h and Δ_h. The fourth and last terms cancel out, since both discrete operators are skew-adjoint. The third term is always negative because the discrete Laplace operator is negative definite. Consequently, the total energy decreases such that the scheme is unconditionally energy stable. $\qquad\square$

3.2. Modified projection method

To avoid the necessity to solve a coupled linear system of equations in dealing with Eqs. (3.3)–(3.7) or Eqs. (3.10)–(3.14), we can modify the BE and CN schemes slightly. The key idea is to march the time integration of the stretching factor $|\mathbf{X}_s|^{n+1}$ by the intermediate velocity $\mathbf{u}^*$ rather than $\mathbf{u}^{n+1}$, and substitute it into the discrete spreading operator $\mathscr{S}_h^n$ suxh that $|\mathbf{X}_s|^{n+1}$ is no longer among the unknowns in the following schemes..

- Modified BE scheme

$$
\rho\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mu\Delta_h\mathbf{u}^* + \mathscr{S}_h^n[\sigma^n] + \sigma_0\Delta t\mathscr{S}_h^n\mathscr{T}_h^n[\mathbf{u}^*],\ \text{in}\ \Omega_h;
\tag{3.19}
$$

$$
\Delta_h\phi = \frac{1}{\Delta t}\nabla_h\cdot\mathbf{u}^*,\ \text{in}\ \Omega_h;\quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0;
\tag{3.20}
$$

$$
\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi,\ \text{in}\ \Omega_h;
\tag{3.21}
$$

$$
\nabla_h p^{n+1} = \rho\nabla_h\phi - \mu\nabla_h(\nabla_h\cdot\mathbf{u}^*) - \sigma_0\Delta t^2\mathscr{S}_h^n\mathscr{T}_h^n[\nabla_h\phi],\ \text{in}\ \Omega_h;
\tag{3.22}
$$

$$
\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x} - \mathbf{X}_k^n)h^2,\ \text{on}\ \Gamma_h;
\tag{3.23}
$$

$$
\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t\,\mathbf{U}_k^{n+1},\ \text{on}\ \Gamma_h.
\tag{3.24}
$$

- Modified CN scheme

$$\rho \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\mu}{2}\Delta_h(\mathbf{u}^* + \mathbf{u}^n) + \mathscr{S}_h^n[\sigma^n]$$

$$+ \frac{\sigma_0\Delta t}{4}\mathscr{S}_h^n\mathscr{T}_h^n[\widehat{\mathbf{u}}^n] + \frac{\sigma_0\Delta t}{4}\mathscr{S}_h^n\mathscr{T}_h^n[\mathbf{u}^*] , \;\; \text{in } \Omega_h ; \tag{3.25}$$

$$\Delta_h\phi = \frac{1}{\Delta t}\nabla_h \cdot \mathbf{u}^* , \;\; \text{in } \Omega_h ; \quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0 ; \tag{3.26}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi , \;\; \text{in } \Omega_h ; \tag{3.27}$$

$$\nabla_h p^{n+1/2} = \rho\nabla_h\phi - \frac{\mu}{2}\nabla_h(\nabla_h \cdot \mathbf{u}^*) - \frac{\sigma_0\Delta t^2}{4}\mathscr{S}_h^n\mathscr{T}_h^n[\nabla_h\phi] , \;\; \text{in } \Omega_h ; \tag{3.28}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x} - \mathbf{X}_k^n)h^2, \quad \widehat{\mathbf{U}}_k^n = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^n\delta_h(\mathbf{x} - \mathbf{X}_k^n)h^2 , \;\; \text{on } \Gamma_h ; \tag{3.29}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t(\mathbf{U}_k^{n+1} + \widehat{\mathbf{U}}_k^n)/2 , \;\; \text{on } \Gamma_h . \tag{3.30}$$

The lower computational cost of the above numerical schemes is of interest. As we mentioned before, the discrete operators $\mathscr{S}_h^n$ and $\mathscr{T}_h^n$ are skew-adjoint with matrix forms related by $\mathscr{T}_h^n = -(h^2/\Delta s)(\mathscr{S}_h^n)^T$. Thus the resultant matrix equations for $\mathbf{u}^*$ in Eq. (3.19) and Eq. (3.25) are both symmetric and positive definite, and we can solve efficiently using the aggregation-based multigrid (AGMG) method recently developed by Notay [25]. In Eqs. (3.20) and (3.26), the Poisson equation for pressure increment ϕ can be solved by the public software package FISHPACK [1]. Despite the fact that these modified BE and CN schemes are not exactly unconditionally energy stable (unlike the original schemes), it is notable that they benefit from the stability such that the time step size Δt can be tremendously alleviated, as our numerical results below show.

4. Application to Simulating Vesicle Dynamics

Kim & Lai [13] developed a penalty IB method to study the dynamics of inextensible vesicles by introducing a dual representation $\mathbf{X}(s,t)$ and $\mathbf{Y}(s,t)$ of the immersed boundary, where the first representation $\mathbf{X}(s,t)$ interacts with the fluid directly as in traditional IB computation and the second $\mathbf{Y}(s,t)$ follows the equations of vesicle dynamics (including the inextensibility constraint) without direct interaction with the fluid dynamics. Both IB representations are linked by stiff springs. The advantage of this penalty idea is that the solution of fluid and vesicle dynamics is decoupled at each time step, so the traditional IB implementation can be applied without much extra effort. Nevertheless, the tension must be solved in this approach, so the penalty numbers has to be chosen sufficiently large to keep the two IB representations close enough, and the time step size must also be small in their explicit forcing computation. Recently, we simplified the penalty approach by introducing a spring-like tension to keep the vesicle boundary nearly inextensible. As a result, the tension (Lagrange multiplier for inextensibility) no longer needs to be found as part of the solution, and we applied this simplified penalty approach to simulate the dynamics of

three-dimensional axisymmetric vesicles in Navier-Stokes flows [12], but the time step size had to be sufficiently small because the IB force was computed at the beginning of each time step. Here we use our two semi-implicit modified schemes given above that permit much larger time steps, to simulate the vesicle dynamics in Navier-Stokes flows.

Vesicles are closed lipid membranes suspended in a viscous fluid. The membrane force consists of elastic and bending parts, according to

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) - c_b \frac{\partial^4 \mathbf{X}}{\partial s^4} \,, \tag{4.1}$$

where c_b is the bending rigidity. The tension σ introduced here is an unknown function, acting as the Lagrange multiplier for enforcing the local inextensibility of the membrane given by

$$\nabla_s \cdot \mathbf{U} = \frac{\partial \mathbf{U}}{\partial \boldsymbol{\tau}} \cdot \boldsymbol{\tau} = 0 \,, \ \text{on } \Gamma \,. \tag{4.2}$$

As in Refs. [2,12], we replace the unknown tension σ by

$$\sigma = \sigma_0 \left(|\mathbf{X}_s| - |\mathbf{X}_s|^0 \right) \,, \tag{4.3}$$

where the elastic coefficient $\sigma_0 \gg 1$ and $|\mathbf{X}_s|^0$ is the local stretching factor of the initial vesicle configuration. This spring-like tension is intended to keep the stretching factor $|\mathbf{X}_s|$ close to its initial counterpart $|\mathbf{X}_s|^0$. (Note that exact inextensibility means $|\mathbf{X}_s| = |\mathbf{X}_s|^0$ for all time.) The numerical schemes presented in Section 3 can be extended straightforwardly to handle the tension computation, by replacing the constant rest length r_0 with the initial stretching factor $|\mathbf{X}_s|^0$. We leave the computational detail for the bending force to the Appendix.

Unless otherwise stated, throughout this section we set the fluid density $\rho = 1$, the viscosity $\mu = 1$, and the computational domain $\Omega = [-1,1] \times [-1,1]$. The stopping tolerance of the iterative method AGMG is 10^{-6}. All numerical runs were carried out on a PC with 16G RAM using double precision arithmetic.

4.1. Convergence study

The convergence of our modified BE and CN schemes is first investigated. We put a vesicle with an elliptical shape $\mathbf{X}(s,0) = (0.2\cos(s), 0.5\sin(s))$ in quiescent flow initially. The elastic coefficient is chosen to be $\sigma_0 = 10^5$, and the bending coefficient $c_b = 0.01$. We use different mesh sizes where $m = n = 64, 128, 256, 512$ with the corresponding mesh $h = 2/m$. We also set the Lagrangian mesh width as $\Delta s \approx h/2$, and the time step $\Delta t = h/4$. Since the fluid is incompressible and the vesicle boundary is inextensible, the enclosed area and the total perimeter of the vesicle should be conserved as time evolves. For the fluid variables, we choose the result obtained from the finest mesh $m = 512$ as our reference solution, and compute the maximal error between this reference solution and the numerical solution. All numerical solutions were computed up to time $T = 0.125$.

We list the results obtained by modified BE and CN schemes in Table 1, showing the relative errors of the vesicle area and the perimeter, the maximum errors of the vesicle

Table 1: The mesh refinement results for the vesicle area A_h, vesicle perimeter L_h, boundary configuration $\mathbf{X}_h$, and velocity components u_h and v_h.

BE	$m=n=64$	$m=n=128$	rate	$m=n=256$	rate
$\lvert A_h-A_0\rvert/A_0$	1.470e-3	1.252e-3	0.23	7.162e-4	0.81
$\lvert L_h-L_0\rvert/L_0$	7.444e-3	2.684e-3	1.47	1.154e-4	1.22
$\lVert\mathbf{X}_h-\mathbf{X}_{\text{ref}}\rVert_\infty$	4.209e-3	8.477e-4	2.31	4.762e-4	0.83
$\lVert u_h-u_{\text{ref}}\rVert_\infty$	5.453e-2	8.110e-3	2.75	1.578e-3	2.36
$\lVert v_h-v_{\text{ref}}\rVert_\infty$	5.628e-2	1.098e-2	2.36	3.203e-3	1.78
CN	$m=n=64$	$m=n=128$	rate	$m=n=256$	rate
$\lvert A_h-A_0\rvert/A_0$	2.688e-3	2.082e-3	0.37	1.148e-4	0.86
$\lvert L_h-L_0\rvert/L_0$	6.899e-3	2.447e-3	1.50	1.052e-4	1.22
$\lVert\mathbf{X}_h-\mathbf{X}_{\text{ref}}\rVert_\infty$	3.296e-3	9.643e-4	1.77	6.821e-4	0.50
$\lVert u_h-u_{\text{ref}}\rVert_\infty$	7.318e-2	4.919e-2	0.57	2.744e-2	0.84
$\lVert v_h-v_{\text{ref}}\rVert_\infty$	7.561e-2	5.790e-2	0.39	1.935e-2	1.58

boundary configuration, and the fluid velocity field. Since the fluid variables are defined at the staggered grid, the numerical solution on a refined mesh does not coincide with the one obtained from a coarser mesh, and we simply used linear interpolation to compute the solutions at the same grid locations. From Table 1, we note that the numerical results show roughly first-order convergence for all solution variables.

4.2. Maximal time step comparison

Let us now investigate the numerical stability of the present schemes, by testing the maximal time steps for the earlier explicit scheme (EP) versus our modified BE and CN schemes. We used three different elastic coefficients $\sigma_0 = 10^7, 10^8, 10^9$ to study the numerical stability, by comparing the maximal time step that can be used in each scheme. To determine the maximal time step, we ensured the vesicle boundary behaviours are reasonable, by requiring the relative errors of both the area and perimeter to be within 1% and that there is no numerical instability. The numerical parameters were the same as those in the previous convergence study, except that the initial vesicle configuration was chosen to be $\mathbf{X}(s) = (0.1\sin(s), 0.5\cos(s))$. Table 2 shows the maximal time steps for the three schemes, and it can be seen that the time step for both of the modified BE and CN schemes can be chosen to be 3 to 4 orders larger than the time step in the explicit scheme. Moreover, as the grid becomes finer or the elastic coefficient σ_0 becomes larger, the time step must become smaller to maintain numerical stability in the explicit scheme. However, for both BE or CN schemes we can always set $\Delta t = O(h)$ to maintain the desired numerical stability.

Table 3 shows the average CPU time (in seconds) for each time step, and the total CPU time for the computation up to $T = 1$. The modified BE and CN schemes clearly outperform the explicit scheme in terms of the total CPU time.

Table 2: Maximum time steps for the explicit(EP) and the modified BE and CN schemes, for various elastic coefficients σ_0.

m, n	$\sigma_0 = 10^7$			$\sigma_0 = 10^8$			$\sigma_0 = 10^9$		
	EP	BE	CN	EP	BE	CN	EP	BE	CN
128	2.22e-5	1.56e-2	1.56e-2	4.36e-6	1.56e-2	1.56e-2	1.02e-6	1.56e-2	1.56e-2
256	1.03e-5	7.81e-3	7.81e-3	1.74e-6	7.81e-3	7.81e-3	3.81e-7	7.81e-3	7.81e-3
512	4.74e-6	3.91e-3	3.91e-3	7.44e-7	3.91e-3	3.91e-3	1.53e-7	3.91e-3	3.91e-3

Table 3: The average CPU time (in seconds) of each time step for different schemes. The total time with "*" means the estimated value.

m, n	EP			BE			CN		
	Δt	Average	Total	Δt	Average	Total	Δt	Average	Total
128	1.02e-6	0.07	68627	1.56e-2	0.13	8	1.56e-2	0.13	8
256	3.81e-7	0.16	419947*	7.81e-3	0.39	50	7.81e-3	0.43	55
512	1.53e-7	0.45	2941176*	3.91e-3	1.25	320	3.91e-3	1.45	371

4.3. Tank-treading motion under shear flow

In a more physically realistic application, we consider the transient motion of a vesicle with the initial configuration $\mathbf{X}(s) = (0.1\sin(s), 0.5\cos(s))$ under shear flow $\mathbf{u} = (\gamma y, 0)$. The elastic coefficient is chosen to be $\sigma_0 = 10^5$, and the bending coefficient $c_b = 0.01$. It is well-known that the equilibrium dynamics of a vesicle under a simple shear flow undergoes tanking-treading motion if the viscosity contrast is under a certain threshold [14], where by tank-treading motion we mean that the configuration of the vesicle remains stationary while there is a tangential motion along the vesicle boundary. Fig. 2 shows the evolutional motion of the vesicle at different times. One can see that after some time the vesicle shape does not change, although the Lagrangian point (marked by $*$) along the interface moves with its tangential velocity. This tank-treading motion shown is in good agreement with previous studies [13, 20, 31, 33].

As discussed elsewhere [13, 15, 16, 31, 33], the motion of a steady vesicle under shear flow can be characterised by the inclination angle θ between the long axis of the vesicle and the flow direction and the tank-treading frequency $f = 2\pi / \int_\Gamma dl / u_\tau$, where u_τ is the tangential velocity component. The inclination angle has been found to depend strongly on the reduced area $V = 4\pi A_0 / L_0^2$, where A_0 is the vesicle area and L_0 is the total perimeter of vesicle. (Under this definition, a circle has the reduced area $V = 1$, while an ellipse with larger aspect ratio has a smaller reduced area.) As the reduced area increases the inclination angle also increases, but the angle is independent of the shear rate γ. This behaviour is verified in the left panel of Fig. 3, which shows the steady inclination angle (θ/π) versus the reduced area (V) with different shear rates $\gamma = 1, 5, 10$. Our numerical results are again in good agreement with those obtained in previous studies [13, 31].

The right panel of Fig. 3 shows the tank-treading frequency f versus the reduced area V for different shear rates. It is evident that as the shear rate increases the tangential
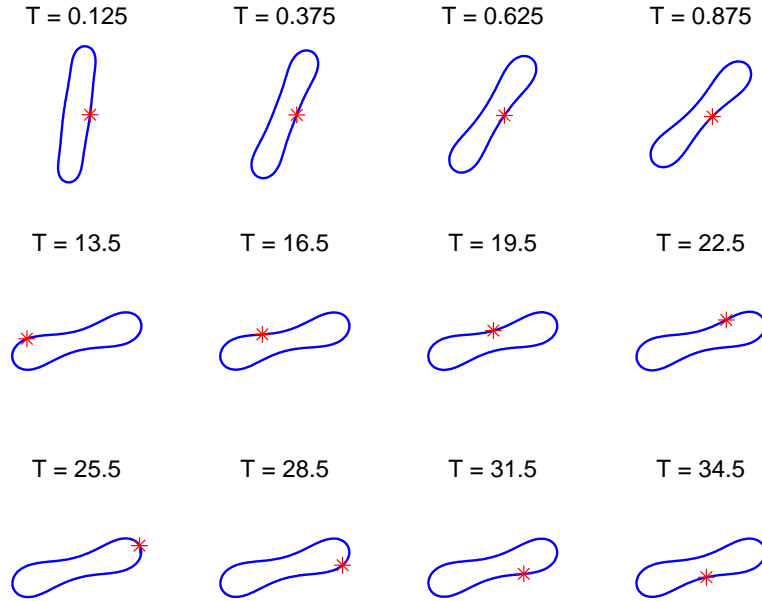
Figure 2: The tank-treading motion of a vesicle under shear flow.

motion becomes stronger, so the frequency becomes larger. Moreover, on fixing the shear rate, if the vesicle has larger reduced area then it has larger frequency as well. Again, our numerical results are in good agreement with those obtained in previous studies [13, 31].

5. Conclusions and Future Work

We have introduced a semi-implicit boundary forcing approach, where the updated stretching factor is used in the forcing term so that the developed backward Euler and Crank-Nicholson schemes are unconditionally energy stable. By using the projection method to solve the fluid equations, our method decouples the pressure and produces a symmetric positive definite system that can be solved efficiently. The resulting modified method therefore benefits from improved numerical stability, so that the time step size can be significantly increased in contrast to the severe time step restriction for explicit boundary forcing schemes. As an application, we use the developed schemes to simulate vesicle dynamics in Navier-Stokes flow. The extension of our new method to axisymmetric 3D and other 3D immersed boundary problems is currently under investigation.
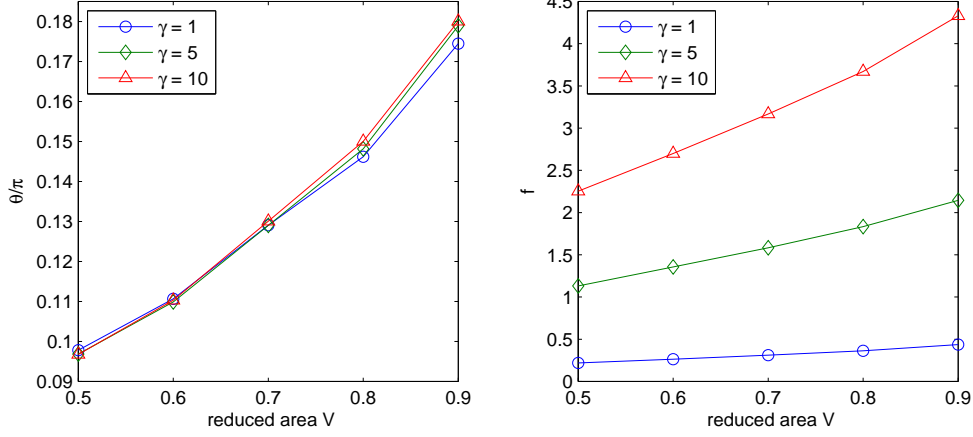
Acknowledgments

Figure 3: The inclination angles θ/π (left) and the tank-treading frequency $f = 2\pi/\int_\Gamma \frac{\mathrm{d}l}{\mathbf{u} \cdot \boldsymbol{\tau}}$ (right) versus reduced areas V with different shear rates, for the tank-treading motion of an inextensible vesicle in a shear flow.

Appendix

This appendix provides the numerical detail on how we compute the bending force in Eq. (4.1). We first define the discrete force spreading operator $\mathscr{F}_h^n$ and the velocity interpolating operator $\mathscr{I}_h^n$ by

$$\mathscr{F}_h^n[\mathbf{F}](\mathbf{x}) = \sum_{k=1}^{M} \mathbf{F}(s_k)\delta_h(\mathbf{x} - \mathbf{X}_k^n)\Delta s \,, \quad \mathscr{I}_h^n[\mathbf{u}](s_k) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}_k^n)h^2 \,, \quad (A.1)$$

respectively. It is well-known $[23, 27]$ that the above both operators are adjoint with each other — i.e.

$$\langle \mathscr{F}_h^n[\mathbf{F}], \mathbf{u} \rangle_{\Omega_h} = \sum_{\mathbf{x}} \left(\sum_{k=1}^{M} \mathbf{F}(s_k)\delta_h(\mathbf{x} - \mathbf{X}_k^n)\Delta s \right) \cdot \mathbf{u}(\mathbf{x})h^2$$

$$= \sum_{k=1}^{M} \mathbf{F}(s_k) \cdot \left(\sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}_k^n)h^2 \right) \Delta s = \langle \mathbf{F}, \mathscr{I}_h^n[\mathbf{u}] \rangle_{\Gamma_h} \,.$$

The singular immersed boundary force arising from the bending is written as

$$\mathbf{f}_b(\mathbf{x}) = -c_b \int_\Gamma \frac{\partial^4 \mathbf{X}}{\partial s^4} \delta(\mathbf{x} - \mathbf{X}(s, t)) \, \mathrm{d}s \,. \quad (A.2)$$

To simplify our notation, we define the discrete fourth-order centred difference operator $\mathscr{A}_h$ as an approximation to the fourth-order derivative. The discretisation for Eq. (A.2) can thus be written as

$$\mathbf{f}_b^{n+1}(\mathbf{x}) = -c_b \mathscr{F}_h^n[\mathscr{A}_h \mathbf{X}^{n+1}](\mathbf{x}) \,.$$

By substituting $\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \mathscr{I}_h^n[\mathbf{u}^{n+1}]$ into the above equation, we have

$$\begin{aligned}\mathbf{f}_b^{n+1}(\mathbf{x}) &= -c_b \mathscr{F}_h^n[\mathscr{A}_h \mathbf{X}^n + \Delta t \mathscr{A}_h \mathscr{I}_h^n[\mathbf{u}^{n+1}]](\mathbf{x}) \\ &= -c_b \mathscr{F}_h^n[\mathscr{A}_h \mathbf{X}^n](\mathbf{x}) - c_b \Delta t \mathscr{F}_h^n \mathscr{A}_h \mathscr{I}_h^n[\mathbf{u}^{n+1}](\mathbf{x}).\end{aligned} \tag{A.3}$$

Since the discrete operators $\mathscr{F}_h^n$ and $\mathscr{I}_h^n$ are self-adjoint and the discrete fourth differential operator $\mathscr{A}_h$ is symmetric positive definite, the above composite operator $\mathscr{F}_h^n \mathscr{A}_h \mathscr{I}_h^n$ is also symmetric positive definite. Thus in our modified BE scheme we need only add the additional singular force

$$\mathbf{f}_b^*(\mathbf{x}) = -c_b \mathscr{F}_h^n[\mathscr{A}_h \mathbf{X}^n](\mathbf{x}) - c_b \Delta t \mathscr{F}_h^n \mathscr{A}_h \mathscr{I}_h^n[\mathbf{u}^*](\mathbf{x}). \tag{A.4}$$

It is notable that the symmetric positive definite matrix structure for $\mathbf{u}^*$ in Eq. (3.19) does not change at all, even when we add this extra bending force term. The bending term for modified CN scheme can be derived similarly, so we omit that detail.

References

[1] J. Adams, P. Swarztrauber and R. Sweet, *Fishpack – A Package of Fortran Subprograms for the Solution of Separable Elliptic Partial Differential Equations*, (1980).

[2] J. T. Beale, *Partially implicit motion of a sharp interface in Navier-Stokes flow*, J. Comput. Phys. **231**, 6159-6172 (2012).

[3] H. D. Ceniceros, J. E. Fisher and A. M. Roma, *Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method*, J. Comput. Phys. **228**, 7137-7158 (2009).

[4] H. D. Ceniceros and J. E. Fisher, *A fast, robust and non-stiff immersed boundary method*, J. Comput. Phys. **230**, 5133-5153 (2011).

[5] J. L. Guermond, P. Minev and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg. **195**, 6011-6045 (2006).

[6] B. E. Griffith and C. S. Peskin, *On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems*, J. Comput. Phys. **208**, 75-105 (2005).

[7] B. E. Griffith, R. D. Hornung, D. M. McQueen and C. S. Peskin, *An adaptive, formally second order accurate version of the immersed boundary method*, J. Comput. Phys. **223**, 10-49 (2007).

[8] R. D. Guy and B. Philip, *A multigrid method for a model of the implicit immersed boundary equations*, Commun. Comput. Phys. **12**, 378-400 (2012).

[9] F. H. Harlow and J. E. Welsh, *Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface*, Phys. Fluids **8**, 2181-2189 (1965).

[10] T.Y. Hou and Z. Shi, *Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations*, J. Comput. Phys. **227**, 9138-9169 (2008).

[11] T. Y. Hou and Z. Shi, *An efficient semi-implicit immersed boundary method for the Navier-Stokes equations*, J. Comput. Phys. **227**, 8968-8991 (2008).

[12] W.-F. Hu, Y. Kim and M.-C. Lai, *An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier-Stokes flows*, submitted for publication.

[13] Y. Kim and M.-C. Lai, *Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method*, J. Comput. Phys. **229**, 4840-4853 (2010).

[14] S. R. Keller and R. Skalak, *Motion of a tank-treading ellipsoldal particle in a shear flow*, J. Fluid Mech. **120**, 27-47 (1982).

[15] V. Kantsler and V. Steinberg, *Orientation and dynamics of a vesicle in tank-treading motion in shear flow*, Phys. Rev. Lett. **95**, 258101 (2005).

[16] M. Kraus, W. Wintz, U. Seifert and R. Lipowsky, *Fluid vesicles in shear flow*, Phys. Rev. Lett. **77**, 3685-3688 (1996).

[17] M.-C. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, J. Comput. Phys. **160**, 705-719 (2000).

[18] M.-C. Lai, Y.-H. Tseng and H. Huang, *An immersed boundary method for interfacial flow with insoluble surfactant*, J. Comput. Phys. **227**, 7279-7293 (2008).

[19] M.-C. Lai, W.-F. Hu and W.-W. Lin, *A fractional step immersed boundary method for Stokes flow with an inextensible interface enclosing a solid particle*, SIAM J. Sci. Comput. **34**, B692-B710 (2012).

[20] Z. Li and M.-C. Lai, *New finite difference methods based on IIM for inextensible interfaces in incompressible flows*, East Asian J. Appl. Math. **1**, 155-171 (2011).

[21] A. A. Mayo and C. S. Peskin, *An implicit numerical method for fluid dynamics problems with immersed elastic boundaries*, Contemp. Math. **141**, 261-277 (1993).

[22] Y. Mori and C. S. Peskin, *Implicit second-order immersed boundary methods with boundary mass*, Comput. Methods Appl. Mech. Engrg. **197**, 2049-2067 (2008).

[23] E. P. Newren, A. L. Fogelson, R. D. Guy and R. M. Kirby, *Unconditionally stable discretizations of the immersed boundary equations*, J. Comput. Phys. **222**, 702-719 (2007).

[24] E. P. Newren, A. L. Fogelson, R. D. Guy and R. M. Kirby, *A comparison of implicit solvers for the immersed boundary equations*, Comput. Methods Appl. Mech. Engrg. **197**, 2290-2304 (2008).

[25] Y. Notay, *An aggregation-based algebraic multigrid method*, Electronic Transactions on Numerical Analysis **37**, 123-146 (2010).

[26] C. S. Peskin, *Flow patterns around heart valves: A numerical method*, J. Comput. Phys. **10**, 220-252 (1972).

[27] C. S. Peskin, *The immersed boundary method*, Acta Numerica **11**, 1-39 (2002).

[28] A. M. Roma, C. S. Peskin and M. J. Berger, *An adaptive version of the immersed boundary method*, J. Comput. Phys. **153**, (1999), 509-534.

[29] J. M. Stockie and B. R. Wetton, *Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes*, J. Comput. Phys. **154**, 41-64 (1999).

[30] C. Tu and C. S. Peskin, *Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods*, SIAM J. Sci. Stat. Comput. **13**, 1361-1376 (1992).

[31] S. K. Veerapaneni, D. Gueyffier, D. Zorin and G. Biros, *A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D*, J. Comput. Phys. **228**, 2334-2353 (2009).

[32] X. Yang, X. Zhang, Z. Li and G.-W. He, *A smoothing technique for discrete delta functions with application to immersed bounary method in moving boundary simulations*, J. Comput. Phys. **228**, 7821-7836 (2009).

[33] H. Zhou and C. Pozrikidis, *Deformation of liquid capsules with incompressible interfaces in simple shear flow*, J. Fluid Mech. **283**, 175-200 (1995).